Electronic Version

Stylesheet Version v1.1.1

Description

[A METHOD FOR CONSIDERING HIERARCHICAL PREEMPTIVE DEMAND PRIORITIES IN A SUPPLY CHAIN OPTIMIZATION MODEL]

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application is related to pending U.S. Patent Application 10/______, filed concurrently herewith to Denton et al., entitled " A METHOD FOR SUPPLY CHAIN COMPRESSION" having (IBM) Docket No. BUR920030197US1; U.S. Patent Application 10/______, filed concurrently herewith to Denton et al., entitled "METHOD FOR PURCHASE ORDER RESCHEDULING IN A LINEAR PROGRAM" having (IBM) Docket No. BUR92004009US1; U.S. Patent Application 10/______, filed concurrently herewith to Denton et al., entitled " A METHOD FOR SUPPLY CHAIN DECOMPOSITION" having (IBM) Docket No. BUR920040007US1; U.S. Patent Application 10/______, filed concurrently herewith

to Denton et al., entitled "A METHOD FOR OPTIMIZING FOUNDRY CAPACITY" having (IBM) Docket No. BUR920030195US1; U.S. Patent Application 10/______, filed concurrently herewith to Denton et al., entitled "METHOD FOR FAIR SHARING LIMITED RESOURCES BETWEEN MULTIPLE CUSTOMERS" having (IBM) Docket No. BUR920040010US1; U.S. Patent Application 10/______, filed concurrently herewith to Denton et al., entitled " METHOD FOR SIMULTANEOUSLY CONSIDERING CUSTOMER COMMIT DATES AND CUSTOMER REQUEST DATES" having (IBM) Docket No. BUR920040008US1; and U.S. Patent Application 10/______, filed concurrently herewith to Orzell et al., entitled "METHOD FOR IDENTIFYING PRODUCT ASSETS IN A SUPPLY CHAIN USED TO SATISFY MULTIPLE CUSTOMER DEMANDS" having Docket No. BUR820030346US1. The foregoing applications are assigned to the present assignee, and are all incorporated herein by reference.

BACKGROUND OF INVENTION

[0002] *Field of the Invention*

[0003] The present invention relates to computer implementable decision support systems for determining solutions to supply chain optimization problems in which customer

demands have associated preemptive priorities to denote the relative importance of satisfying the demand. General methodologies within this field of study include advanced planning systems, optimization and heuristic based algorithms, constraint based programming, and simulation.

[0004] *Description of the Related Art*

[0005] A fundamental problem faced in all manufacturing industries is the allocation of material and capacity assets to meet end customer demand. Production lead times necessitate the advance planning of production starts, interplant shipments, and material substitutions throughout the supply chain so that these decisions are coordinated with the end customers" demand for any of a wide range of finished products (typically on the order of thousands in semiconductor manufacturing). Such advance planning depends upon the availability of finite resources which include: finished goods inventory, work in process inventory (WIP) at various stages of the manufacturing system, and work-center capacity. Often, there are alternative possibilities for satisfying the demand. Products may be built at alternative locations and within a location there may be choices as to which materials or capacity to use to build the product. The product may be built directly or acquired

through material substitution or purchase. When limited resources prevent the satisfaction of all demands, decisions need to be made as to which demand to satisfy and how to satisfy it. This resource allocation problem is often addressed through linear programming.

[0006] The below-referenced U.S. Patents disclose embodiments that were satisfactory for the purposes for which they were intended. The disclosures of both the below-referenced prior U.S. Patents, in their entireties, are hereby expressly incorporated by reference into the present invention for purposes including, but not limited to, indicating the background of the present invention and illustrating the state of the art: U.S. Patent 5,971,585, "Best can do matching of assets with demand in microelectronics manufacturing," October 26, 1999; U.S. Patent 5,943,484, "Advanced material requirements planning in microelectronics manufacturing," August 24, 1999; and Nemhauser, G.L. and Wolsey, L.A., 1999, Wiley, *Integer and Combinatorial Optimization*. Also incorporated for reference into the present invention for purposes including, but not limited to, indicating the background of the present invention and illustrating the state of the art: (Leachman, R. C., R. F. Benson, C. Liu, D. J. Raar. 1996. IMPReSS: An au-

tomated production-planning and delivery-quotation system at Harris Corporation-semiconductor sector. *Interfaces* 26(1) 637.)

SUMMARY OF INVENTION

[0007] The invention provides a method and system for determining a production plan that allocates resources to different demands having priorities by iteratively solving mathematical linear programs. Each mathematical linear program optimizes according to one of a plurality of sets of priorities and each iterative solution is consistent with the previous set of priorities. These priorities are hierarchical and comprise two or more levels of hierarchy.

[0008] Backorder costs penalties are determined independently for each set of priorities and comprise a full spectrum range within each set of priorities. The mathematical linear programs solved in each iteration use the solution to the previous mathematical linear program as a starting point. The invention also adds constraints to the mathematical linear programs at each iteration to ensure that solutions to subsequent iterations are consistent with previous priorities. This method uses a different mathematical linear program for each iteration. The invention solves the mathematical linear programs for higher priori-

ties before solving for lower priorities.

[0009] More specifically, the invention provides a method of allocating resources to a hierarchy of demand priorities in a linear programming production planning system. In particular, the invention aggregates the demand priorities into different priority groups and allocates the resources to the highest priority group of demand priorities using a first linear programming model. Next, the invention allocates remaining resources to the next highest priority group of demand priorities using a second linear programming model. The second linear programming model uses results from the first linear programming model. The invention continues this process by iteratively repeating the process of allocating remaining resources to the remaining groups of demand priorities, in order of priority.

[0010] When repeating the process of allocating remaining resources, the method uses a different linear programming model for each iteration and each different linear programming model uses results of the previous linear programming model. During the allocating processes, each linear programming model fixes variables for all demand priorities except for demand priorities of the priority group to which the resources are currently being allo-

cated. Also, during the allocating processes, each linear programming model allocates the range of backorder costs within the priority group to which the resources are currently being allocated. The invention can also divide the priority groups into different sub-priority tiers. These sub-priority tiers can be processed simultaneously or separately.

[0011] The present invention is a hybrid method which handles hierarchical demand priorities via demand class aggregation and subsequent demand tier disaggregation.The inventive approach is based on aggregating demand priorities into groups, in which each group corresponds to an LP run. The invention generates backorder cost penalties associated with demand priorities within a group to model the consideration of multiple demand classes within a demand class group in the LP model. Further, the invention method iteratively modifies and solves the respective LPs in memory and leverages the results from one solution to the next.

[0012] The inventive method results in substantial computational savings over conventional systems while at the same time respecting demand classes throughout the supply chain. The invention solves relaxed (e.g., modified) versions of

the LP model at each iteration that allow for flexibility in realigning resources (e.g. work-center capacity) to accommodate lower priority groups without sacrificing higher priority groups. Further, the invention can be applied to hierarchical demand priorities by selectively disaggregating demand tiers associated with demand classes.

Brief Description of Drawings

[0013] Figure 1: Overview of the structure of a linear programming application.

[0014] Figure 2: Illustration of backorder coefficients within a priority group.

[0015] Figure 3: Illustration of demand priority aggregation.

[0016] Figure 4: Illustration of hierarchical demand priority based on demand class and demand tier.

[0017] Figure 5: Illustration of the steps of the method.

[0018] Figure 6: Flowchart illustrating one embodiment of the invention.

Detailed Description

[0019] Many factors are used to determine the relative importance of the variety of customer demands that are serviced by a manufacturing firm (e.g. gross margin, strate-

gic importance, forecast vs. committed orders, demand mix considerations etc.). Typically, the implementation of supply-chain planning methods requires that these be translated into a "demand priority" (e.g. 1, 2, 3,...) so that the many customer demands (measured in tens of thousands for division central runs) can be rank ordered according to importance. It is necessary to model this relative importance of demands when trading off the allocation of limited resources. For instance, photolithography tools are typically a production bottleneck in semiconductor manufacturing. When insufficient capacity exists to schedule production for all demands it is expected that work-in-process (WIP) inventory associated with higher priority demands will be processed before competing lower priority WIP.

[0020] It is important to break out the demand priority into a hierarchical set of attributes. For instance, it is common for a manufacturing firm to have two attributes a) demand class and b) demand tier which are used to determine the ultimate demand priority. Figure 4 illustrates this hierarchical structure with an example in which there are four demand classes and each of the demand classes has three associated tiers. Thus, demand class is the most impor-

tant factor and, within a particular demand class, importance is dictated by demand tier. The significance of the hierarchical structure is to denote that demand class is the most important factor dictating priority while demand tier is secondary.

[0021] To contrast the present invention, a conventional production planning linear program "LP" is shown below (such as that described in U.S. Patent 5,971,585, which is incorporated herein by reference). This LP makes decisions including: production starts, material substitutions, and shipments planned to customers, between manufacturing and distribution locations, and from vendor suppliers. A LP is composed of an objective function that defines a measure of the quality of a given solution, and a set of linear constraints. The types of equations used in production planning models are well know to those practiced in the art and include:(1) Material Balance Constraints, which ensure conservation of material flow through the network of stocking points comprising the supply chain.

[0022] (2) Capacity Constraints, which ensure that the capacity available for manufacturing activities is not exceeded.

[0023] (3) Backorder Conservation Constraints, which balance the quantity of a given part backordered in a given planning

period with the quantity backordered in the previous planning period and the net of new demand and new shipments.

[0024] (4) Sourcing Constraints, which define target ranges (minimum and maximum) of shipments that should be made from a particular manufacturing or vendor location in the supply chain.

[0025] A conventional LP formulation is provided below in the form familiar to those practiced in the art; i.e., definition of subscripts, definition of objective function coefficients, definition of constants, definition of decision variables, LP formulation or equations.

[0026] *Definition of Subscripts*

[0027] j – time period

[0028] m – material (part number)

[0029] a – plant location within the enterprise

[0030] n – material being substituted

[0031] z – group (which represents a family or collection of part numbers)

[0032] e – process (a method of purchasing or manufacturing a material at a plant)

[0033] v - receiving plant location

[0034] k - demand center (i.e., customer location) (Note: the set of customer locations is mutually exclusive from the set of plant locations)

[0035] q - demand class which indicates relative priority. The higher the numerical value of q the lower the relative priority.

[0036] w - resource capacity which could be a machine, labor hour, or other constraint

[0037] u represents a consumer location which refers to an internal plant, external demand center, or toa generic indicator meaning any plant/or demand center

[0038] *Definition of Objective Function Coefficients*

[0039] PRC_{jmae} - cost of releasing one piece of part m during period j at plant a using process e

[0040] $SUBC_{jmna}$ - substitution cost per piece of part number n which is being substituted by part number m during period$j$ at plant a

[0041] TC_{jmav} - transportation cost per piece of part number m leaving plant a during period j which are destined for plant v

[0042] $INVC_{jma}$ - inventory cost of holding one piece of part

number m at the end of period j at a particular plant a

[0043] $DMAXC_{jzau}$ – cost per piece of exceeding the maximum amount of shipments of group z parts from plant a to consuming location(s) u during period j

[0044] $DMINC_{jzau}$ – cost per piece of falling short of the minimum amount of shipments specified for group z parts from plant a to consuming location(s) u during period j

[0045] BOC_{jmkq} – backorder cost of one piece of part m at the end of period j for class q demand at customer location k

[0046] *Definition of Constants*

[0047] $DEMAND_{jmkq}$ – demand requested during time period j for part number m at customer location k for demand class q

[0048] $RECEIPT_{jma}$ – quantity of projected wip and purchase order receipts for part number m expected to be received at plant a during time period$j$

[0049] $CAPACITY_{jaw}$ – Capacity of resource w available at plant a during period j to support production starts

[0050] $CAPREQ_{jmaew}$ – Capacity of resource w required for part number m at plant a for process e during period jQ

[0051] $TYPER_{jmaen}$ – quantity of component m needed per part number n during period j at plant a using process e

[0052] $YIELD_{jmae}$ – output of part number m per piece released or started at plant a during time period j using process e

[0053] $SUBQTY_{jmna}$ – quantity of part number m required to substitute for one piece of part number n at plant a during time period j

[0054] $MAXPCT_{jzau}$ – maximum percentage of total shipments of group z (collection of parts) leaving supplier a during period j to support consumption at consuming location(s) u

[0055] $MINPCT_{jzau}$ – minimum percentage of total shipments of group z (collection of parts) leaving supplier a during period j to support consumption at consuming location(s) u

[0056] CT_{jmae} – Cycle time. The number of periods between the release and completion of part m jobs for releases made using process e at plant a during time period j

[0057] TT_{mav} – transport time for part number m from plant a to plant v

[0058] *Definition of LP Decision Variables*

[0059] I_{jma} – Inventory at the end of period j for part number m at a particular plant a

[0060] P_{jmae} – Production starts of part m during period j at plant a using process e

[0061] L_{jmna} – Quantity of part number n which is being substituted by part number m during period$j$ at plant a

[0062] T_{jmav} – Internal shipments of part number m leaving plant a during period j which are destined for plant v

[0063] F_{jmakq} – Shipments of part number *m* leaving plant *a* during period *j* and satisfying class *q* demand at external customer *k*

[0064] B_{jmkq} – Back orders of part *m* at the end of period *j* for class *q* demand at customer location *k*

[0065] H_{jzu} – Total shipments of group *z* (*z* is a "collection" of parts) leaving suppliers during period *j* to support consumption at consuming location(s) *u*.

[0066] S_{jzau} – Amount by which total shipments of parts in *z* from plant *a* to consuming location(s) *u* during period *j* exceeds the maximum amount specified as desired in the sourcing rules

[0067] G_{jzau} – Amount by which total shipments of group *z* parts from plant *a* to consuming location(s) *u* during period *j* falls short of the minimum amount specified as desired in the sourcing rules

[0068] *LP Equations or Formulation*

[0069] The following minimizes the objective function subject to the constraints shown below.

[0070] *Objective Function:*

[0071] *Minimize:*

$$\sum_j \sum_m \sum_a \sum_e PRC_{jmae} P_{jmae} + \sum_j \sum_m \sum_n \sum_a SUBC_{jmna} L_{jmna} + \sum_j \sum_m \sum_a \sum_v TC_{jmav} T_{jmav} +$$
$$\sum_j \sum_m \sum_a INVC_{jma} I_{jma} + \sum_j \sum_z \sum_a \sum_u DMAXC_{jzau} S_{jzau} + \sum_j \sum_z \sum_a \sum_u DMINC_{jzau} G_{jzau} +$$
$$+ \sum_j \sum_m \sum_k \sum_q BOC_{jmkq} B_{jmkq}$$

[0072] *Subject to:*

[0073] Sourcing Constraints:

$$\mathrm{H}_{jzu} = \sum_{\substack{m \\ \varepsilon z}} \sum_a (T_{jmau} + \sum_q F_{jmauq})$$

$$\sum_{\substack{m \\ \varepsilon z}} (T_{jmau} + \sum_q F_{jmauq}) - S_{jzau} \le MAXPCT_{jzau}\ H_{jzu}$$

$$\sum_{\substack{m \\ \varepsilon z}} (T_{jmau} + \sum_q F_{jmauq}) + G_{jzau} \ge MINPCT_{jzau}\ H_{jzu}$$

[0074] Capacity Constraints:

$$\sum_m \sum_e CAPREQ_{jmaew} P_{jmae} \le CAPACITY_{jaw}$$

[0075] Backorder Constraints:

$$B_{jmkq} = B_{(j-1)mkq} + DEMAND_{jmkq} - \sum_a F_{jmakq}$$

[0076] Material Balance Constraints:

$$I_{jma} = I_{(j-1)ma} + RECEIPT_{jma} + \sum_{\substack{xsi.t \\ x+CT_{xmae}=j}} \sum_e YIELD_{xmae} * P_{xmae} + \sum_n L_{jmna} +$$
$$\sum_{\substack{xs.t \\ x+TT_{mav}=j}} \sum_v T_{xmva} - \sum_n SUBQTY_{jmna} * L_{jmna} - \sum_v T_{jmav} - \sum_k \sum_q F_{jmakq}$$
$$- \sum_{\substack{nst.m \\ isacomponent \\ of\,n}} \sum_e QTYPER_{jmaen} P_{jnae}$$

[0077] Non-Negativity Constraints:

all $X_{i,j,...} \geq 0$, where X is a generic decision variable and *i*, *j* etc. represent generic subscripts

[0078] Other approaches use a heuristic approach to schedule in which WIP is sequenced according to its pegged demand priority at the various work-centers throughout the supply chain. While this can be effective from the point of view of rigorously considering demand priorities, these heuristics typically fail to find globally optimal solutions. From the point of view of global optimization of resource allocation, linear programming (LP) is a superior approach. However, the LP approach relies on cost penalty weightings to "encourage" demand priority compliance and in practice demand priorities may be violated (e.g. a high yield low priority item may be favored over a low yield high priority item).

[0079] The Leachman method (Leachman, R. C., R. F. Benson, C. Liu, D. J. Raar. 1996. IMPReSS: An automated production-planning and delivery-quotation system at Harris Corporation-semiconductor sector. *Interfaces* 26(1) 637) incorporates the consideration of "preemptive" demand priorities into an LP modeling framework in which each LP corresponds to a single demand priority. However, this method can have severe disadvantages in terms of computational efficiency, and does not consider hierarchical

demand prioritizationThe inventive approach, on the other hand, is a method based on aggregating demand priorities into groups, in which each group corresponds to an LP run. (Figure 3 illustrates the grouping of priorities into sets.) Thus, the present invention is a hybrid method which handles hierarchical demand priorities via demand class aggregation and subsequent demand tier disaggregation. The invention generates backorder cost penalties associated with demand priorities within a group to model the consideration of multiple demand classes within a demand class group in the LP model. Further, the inventive method iteratively modifies and solves the respective LPs in memory and leverages the results from one solution to the next.

[0080] The inventive method results in substantial computational savings over conventional systems while at the same time respecting demand classes throughout the supply chain. The invention solves relaxed (e.g., modified) versions of the LP model at each iteration that allow for flexibility in realigning resources (e.g. work-center capacity) to accommodate lower priority groups without sacrificing higher priority groups. Further, the invention can be applied to hierarchical demand priorities by selectively dis-

aggregating demand tiers associated with demand classes.

[0081] Ideally, the method used for modeling backorder costs would guarantee demand compliance for any number of demand priorities in a single LP run. In theory this could be done by setting the backorder penalty for a particular demand priority arbitrarily higher than the penalty for the next lower demand priority. However, in practice due to finite numerical accuracy of floating point operations there are bounds on the allowable range of objective function penalties (experimental evidence implies this is about .01 to 1,000,000 when operating with double precision using the 64 bit C/C++ compilation mode), and the delta between priorities which is recognizable. Therefore, with the invention backorder costs penalties are determined independently for each set (group) of priorities. Within each group of demand priorities, the backorder penalties comprise a full spectrum range of objective function penalties. Thus, each successive linear programming model allocates a full range of backorder costs within the priority group to which the resources are currently being allocated to avoid the foregoing problem.

[0082] The numerical bounds on backorder cost penalties are il-

lustrated in the example of Figure 2 in which there are lower and upper bounds on the cost penalties based on a recursive set of mathematical equations described below. Figure 2 further illustrates three possibilities for calibrating backorder costs from a continuous range of possibilities. In Figure 2, the examples illustrate costs which are (a) decreasing at an increasing rate with respect to demand class (b) linearly decreasing in demand class and (c) decreasing at a decreasing rate with respect to demand class. The inventive modeling backorder costs is based on the following recursive equations

[0083] Delta_{j+1} = C*Delta_{j}

[0084] where Delta is chosen, such that within each group of priority demands, the most and least important demand priority correspond to the maximum and minimum allowable backorder costs respectively. If there are N demand priorities in the group and we start with the initial condition that backorder cost BC_{1}=max and model the costs as follows

[0085] BC_{1} = MAX_COST

[0086] BC_{2}=Delta*MAX_COST

[0087] BC_{3}=C*Delta^{2}*MAX_COST

[0088] BC_{i} = C^{i-2}*Delta^{i-1}*MAX_COST

[0089] Since we require that BC_{n}=MIN_COST it follows that

[0090] Delta = (MIN_COST/(c^{(n-2)*(n-1)/2}*MAX_COST))^{1/(n-1)}

[0091] The parameter C is a shape parameter (defined by the user) that influences the relative difference between back-order costs from one demand priority to the next. Given the assumption that backorder costs should be decreasing in the importance of the demand priority the values C can take are in the following range

[0092] Lower Bound (LB) = (MIN_COST/MAX_COST)^)^{2.0/((n-2)*(n-1))}

[0093] Upper Bound (UB) = 1/LB

[0094] where LB and UB follow from the requirement that Delta < 1 and C*Delta < 1 respectively. Depending on the choice for C in the range [LB, UB] different relative differences between backorder costs are obtained. These differences are illustrated in Figure 2. In the present implementation of the invention typically C=LB + 2/3*(UB-LB) is chosen to maximize the relative difference between backorder costs from one demand priority to the next with some preference (determined by the choice of C) given to the differ-

ence between more important demand priorities. As C approaches UB the preference for relative differences in demand priority is shifted from less important to more important demand classes.

[0095] The following is a pseudo code description and corresponding flowchart shown in Figure 5 illustrate an implementation of the above described idea. With the invention any commercial linear program solver could be employed.

[0096] Definitions:

[0097] N(i)=number of demand classes in group i

[0098] M = number of demand tiers per demand class

[0099] n=number of demand priority groups

[0100] F_{m,j,a,k,q}= LP decision variable representing quantity of shipments of part m in period j from plant, a, to customer location, v, with priority q

[0101] B_{m,j,k,q} = LP decision variable representing the quantity of part m to backorder in period j for customer k of priority q

[0102] D_{m,j,k,q} = demand for part m in period j from customer k of priority q

[0103] BO = backorder constraints of the form B_{m,j,k,q} = B_{m,j-1,k,q} + D_{m,j,k,q} Sum(over a) F_{m,j,a,k,q}

[0104] BC_{m,j,k,q} = LP cost penalty for backorder variable B_{m,j,k,q}

[0105] MIN_COST = precision based lower bound on BC penalty

[0106] MAX_COST = precision based upper bound on BC penalty

[0107] UB = upper bound

[0108] LB = lower bound

[0109] Step 1: User defines a set of "demand class groups"from the set of demand classes. For example, 10 demand classes could be split into 2 groups [1,...3] and [4,..,10]. The groups are indexed from i = 1, ...n and the number of distinct demand classes in group i is N(i). Set i = 1.

[0110] Step 2: Fix all F_{m,j,a,k,q} variables where demand class q is in a numerically higher (less important) priority group than group i and unfix all variables such that q is in a numerically lower (more important) priority group than i or equal priority as group i. Fixing and unfixing is done by setting variable_upper_bound = 0.0 and variable_upper_bound = INFINITY respectively.

[0111] Step 3: Recalibrate cost penalties for variables, B_{m,j,k,q}, with q in group i. This method is based on above discussion about maximizing the relative difference between each priority in group i subject to user defined cost shape

parameters.

[0112] //Bounds on shape parameters

[0113] LB = (MIN_COST/MAX_COST)^{2.0/((N(i)-2)*(N(i)-1))}

[0114] UB = 1.0/lower_bound

[0115] //Convert user defined parameter, C, from [0,1] range to [LB,UB]

[0116] Factor = C*LB + (1.0- C)*UB

[0117] //Compute relative cost difference factor

[0118] Delta=MIN_COST/(Factor^{(N(i)-2)*(N(i)-1)/2}*MAX_COST))^{1.0/(N(i)-1)})

[0119] //Compute penalties from deltas

[0120] Cost[0] = MAX_COST

[0121] for(j=1;j<N(j);j++) Cost[j] = factor^{j-1}*Delta*Cost[j-1]

[0122] //Set BC"s to computed cost penalties

[0123] For(all BC_{m,j,k,q} such that q an element of group i){

[0124] DP = relative group i demand priority for current BC_{m,j,k,q}

[0125] BC_{m,j,k,q} = Cost[DP]

[0126] }

[0127] Step 4: Solve modified group i LP:

[0128] if(i==1) solve LP using normal OSL simplex LP method

[0129] else warm start LP solution using previous solution as an advanced basis

[0130] Step 5: Add a new constraint set to the current LP model enforcing that B_{m,j,k,q} variables in group i are lower bounded based on the current LP solution from Step 4. These constraints have the form:

[0131] For a particular period, j, and demand class q: Sum(parts m, customer locations k) B_{m,j,k,q} <= Sum(parts m, customer locations k) (B_{m,j,k,q} from Step 4)

[0132] Step 6: Reset the LP basis with current variables fixed using the dual simplex method. Increment group to i=i+1. If (i==n+1) go to step 7, otherwise go to step 2.

[0133] In the remaining description of the method we have two alternative embodiments which we represent as Step 7A and Step 7b:

[0134] Step 7a: For each demand class disaggregate BO constraints and associated variables into separate demand tiers. For example, the BO constraint

[0135] B_{m,j,k,q} = B_{m,j-1,k,q} + D_{m,j,k,q} Sum(over a) F_{m,j,a,k,q}

[0136] would become a set of constraints indexed by tier, t, as follows

[0137] B_{m,j,k,q,t} = B_{m,j-1,k,q,t} + D_{m,j,k,q,t} Sum(over a) F_{m,j,a,k,q,t} , for all t=1,...,M

[0138] Thus, the constraints added in Step 5 are disaggregated. The previous (aggregated) solution can be used to provide a feasible starting solution for reoptimizing the LP with the additional disaggregated constraints. For example, if there were two demand tiers then F_{m,j,a,k,q1} = D1* F_{m,j,a,k,q} and F_{m,j,a,k,q2} = D2* F_{m,j,a,k,q}, where D1 and D2 represent the fraction of demand in period j that is in tier 1 and tier 2 respectively. Once the revised solution to the disaggregated LP is determined, the F and B variables for the current demand class are fixed to their current values, the next demand class is disaggregated, solved, and so on. This is carried our until all demand classes have been considered.

[0139] Step 7b: For each variable, B_{m,j,k,q}, the invention recalculates penalties for all of the demand classes that reflects priority differences within a class. For example rather than add a new demand tier index, t, to variables, B_{m,j,k,q}, instead the associated penalty would be a weighted average of the different demand tiers for cus-

tomers, k, within class, q. A final run with these modified cost penalties results in a solution which:a) respects demand class priorities pre-emptively based on the constraints added in Step 5b) given the myriad of solutions respecting (a), this final solution will be one which tends to respect the differences in demand tierThe alternative Steps 7a and 7b allow for the ability to tradeoff different criteria. For instance, Step 7a is more accurate in the way that it applies demand priorities than Step 7b. On the other hand, Step 7b is likely to be computationally more efficient since it does not involve the disaggregation of variables and constraints that is done in Step 7a. Figure 5 illustrates the flow of the above steps in the method.

[0140] Figure 6 is a more general flowchart showing the operation of the invention. In item 600, the invention aggregates the demand priorities into different priority groups. In item 602, the invention allocates the resources to the highest priority group of demand priorities using a first linear programming model. Next, in decision block 604, the invention determines if there are additional groups of priority demands to process. If so, the invention modifies the linear program to create a second linear programming model as shown in item 606. Then processing returns to

item 602 where the invention allocates remaining resources to the next highest priority group of demand priorities using a second linear programming model. The second linear programming model uses results from the first linear programming model. The invention continues this process by iteratively repeating the process of allocating remaining resources to the remaining groups of demand priorities in order of priority. After all groups are processed (or there are no resources remaining) this portion of the process is complete 608.

[0141] When repeating the process of allocating remaining resources, the method uses a different linear programming model for each iteration and each different linear programming model in item 606 uses results of the previous linear programming model. In each of the linear programming models created in item 606, the invention fixes variables associated with all demand priorities that are in lower priority (numerically higher) groups than the priority group to which the resources are currently being allocated. Also, each linear programming model in item 606 allocates the full range of backorder costs within the single priority group to which the resources are currently being allocated. Therefore, each iteration has the benefit of

a full range of backorder costs. The invention can also divide the priority groups into different sub-priority tiers. These sub-priority tiers can be processed simultaneously or separately.

[0142] The present invention has been implemented on an IBM P690 server using the AIX operating system. The steps for implementing the present invention are preferably programmed in C/C++. It should be understood by those of ordinary skill in the art, however, that the represent invention is not limited to the above implementation and is independent of the computer/system architecture. Accordingly, the present invention may equally be implemented on other computing platforms, programming languages and operating systems, and also may be hardwired into a circuit or other computational component.

[0143] The present invention is a hybrid method which handles hierarchical demand priorities via demand class aggregation and subsequent demand tier disaggregation.The inventive approach is based on aggregating demand priorities into groups, in which each group corresponds to an LP run. The invention generates backorder cost penalties associated with demand priorities within a group to model the consideration of multiple demand classes within a de-

mand class group in the LP model. Further, the invention method iteratively modifies and solves the respective LPs in memory and leverages the results from one solution to the next.

[0144] The inventive method results in substantial computational savings over conventional systems while at the same time respecting demand classes throughout the supply chain. The invention solves relaxed (e.g., modified) versions of the LP model at each iteration that allow for flexibility in realigning resources (e.g. work-center capacity) to accommodate lower priority groups without sacrificing higher priority groups. Further, the invention can be applied to hierarchical demand priorities by selectively disaggregating demand tiers associated with demand classes.

[0145] While the invention has been described in terms of the preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.